TwinPeer: Enhancing Peer to Peer Network Connectivity in Blockchain Systems using Digital Twin

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

YELLAVENI SATHISH KUMAR (MT23MCS015)

under the guidance of

Dr. PRASHANTH PVN

Assistant Professor

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

JUNE 2025

TwinPeer: Enhancing Peer to Peer Network Connectivity in Blockchain Systems using Digital Twin

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

YELLAVENI SATHISH KUMAR (MT23MCS015)

under the guidance of

Dr. PRASHANTH PVN

Assistant Professor

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

JUNE 2025

©Visvesvaraya National Institute of Technology (VNIT) 2025

Department of Computer Science & Engineering
Visvesvaraya National Institute of Technology, Nagpur

Declaration

I, **YELLAVENI SATHISH KUMAR**, hereby declare that this dissertation titled **"Twin-Peer: Enhancing Peer to Peer Network Connectivity in Blockchain Systems using Digital Twin"** is carried out by me in the Department of Computer Science & Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier, whole or in part, for the award of any degree at this or any other Institution.

Name and signature
Date: 17/06/25

Certificate

This to certify that the dissertation titled **"TwinPeer: Enhancing Peer to Peer Network Connectivity in Blockchain Systems using Digital Twin"** is submitted by **YELLAVENI SATHISH KUMAR** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science & Engineering**, VNIT Nagpur. The work is comprehensive, complete, and fit for final evaluation.

17/6/25

Dr. PRASHANTH PVN
Department of Computer Science & Engineering, VNIT Nagpur

Dr. M. P. Kurhekar
Head, Department of Computer Science & Engineering, VNIT Nagpur
Date:

विभाग प्रमुख
Head
संगणक विज्ञान एवं अभियांत्रिकी विभाग
Dept. of Computer Science & Engg.
वि.रा.प्रौ.सं. नागपुर-१० (भारत)
V.N.I.T., Nagpur-440010. (India)

Department of Computer Science & Engineering
Visvesvaraya National Institute of Technology, Nagpur

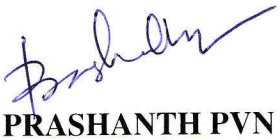
Non-Plagiarism Certificate

Certified that M. Tech Dissertation titled **"TwinPeer: Enhancing Peer to Peer Network Connectivity in Blockchain Systems using Digital Twin"** submitted by **YELLAVENI SATHISH KUMAR**, Enrollment No. **MT23MCS015**, ID No. **30640** has been checked for plagiarism using TURNITIN software and the overlap is found within prescribed limits. A summary of the report is given below.

ORIGINALITY REPORT

5%	2%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

YELLAVENI SATHISH KUMAR
Enrollment No: MT23MCS015



Dr. PRASHANTH PVN
Assistant Professor
Department of Computer Science & Engineering
VNIT, Nagpur

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and sincere appreciation to my supervisor, Dr. PRASHANTH PVN Sir, for offering me the opportunity to work on this project on TwinPeer: Enhancing Peer to Peer Network Connectivity for Blockchain Systems using Digital Twin. His constant support, invaluable guidance, expertise, and encouragement have been instrumental. I am truly fortunate to have the opportunity to work under his mentorship.

I would like to extend my heartfelt thanks to all the teaching and non-teaching faculty members of the Department of Computer Science and Engineering present at Visvesvaraya National Institute of Technology Nagpur for their knowledge sharing and provision of vital resources when necessary.

I would also like to express my gratitude to my family and friends for their unwavering support and encouragement throughout this journey. Their love, motivation, and belief in my abilities have been the driving force behind my accomplishments.

YELLAVENI SATHISH KUMAR

ABSTRACT

Optimized peer connectivity is crucial for reducing network latency in peer-to-peer (P2P) blockchain networks which in-turn contributes to achieve higher transaction rates. A new peer joins the P2P network by connecting to a random set of existing peers making these networks inherently random, resulting in increased network latency. Connecting peers that are geographically proximal (nearest neighbor connectivity (NNC) approach) helps to reduce the network latency, however, it creates network imbalance and leads to formation of hub nodes. Therefore, having a global view of the entire network is essential for making informed decisions about peer connectivity while balancing the network. This paper introduces TwinPeer, a Digital Twin–driven peer connectivity optimization framework designed to enhance propagation efficiency, decentralization, and scalability in blockchain P2P networks. At its core, TwinPeer employs a Digital Twin for blockchain P2P Network' (DTPN) that stores the real time information of the peers including their geo-location, connectivity and latencies. Further, this work proposes 'Proximal and Degree Balanced Connectivity Algorithm (PDBCA)' which leverages DTPN to strategically identify peers based on their geographical proximity and existing connection load for every new peer joining the P2P network. Experimental evaluations demonstrate that PDBCA improves overall network latency while balancing the P2P network compared to Random and NNC approaches.

TABLE OF CONTENTS

Title **Page No.**

ABSTRACT . . . . . **iv**

TABLE OF CONTENTS . . . . . **v**

LIST OF FIGURES . . . . . **vii**

1 INTRODUCTION **2**

1.1 Motivation 2

1.2 Problem Statement 2

1.3 Objectives 3

1.4 Scope of the Study 3

1.5 Thesis Outline 4

2 LITERATURE REVIEW **6**

2.1 Background and Motivation 6

2.1.1 Digital Twin Concepts for Network Management 7

2.2 Related Work 8

2.2.1 Non-Clustered Approaches for Blockchain P2P Networks 8

2.2.2 Clustering approaches for Blockchain P2P Networks 9

2.2.3 Digital Twin related work 10

3 SYSTEM DESIGN **11**

3.1 System Overview 11

3.1.1 Proximal and Degree Balanced Connection Algorithm (PDBCA) 13

3.1.2 Comparison of NNC and PDBCA Algorithms 15
3.1.3 Time Complexity of PDBCA 16

4 IMPLEMENTATION DETAILS 17

5 EXPERIMENTAL RESULTS 19
5.0.1 Comparison of Average Degree 20
5.0.2 Convergence Time comparison 21
5.0.3 Comparison of Message complexity 23

6 CONCLUSION 24

REFERENCES 24

List of Figures

3.1 For a new node F, PDBCA returns the IP addresses of the nearest peers A,B,C. Connections are established with peers A,B,C. 12

3.2 Sample topology formed based on NNC and PDBCA algorithms. Nodes degrees are evenly distributed and hub nodes C and F in NNC are avoided in PDBCA. 16

5.1 Comparison of average degree for different network sizes 20

5.2 Comparison of degree distribution for 20k node network 21

5.3 Comparison of Convergence time 22

5.4 Comparison of Number of Messages Forwarded 23

PUBLICATIONS

PUBLICATION No. 1

Title: Towards Optimized Peer Connectivity in Blockchain Networks using Digital Twin

Authors: Dr.Prashanth PVN, Yellaveni Sathish Kumar

Conference: THE 34th International Telecommunication Networks and Applications Conference (ITNAC)

Location: Sydney, Australia

Details: The paper has been published.

Index: Scopus

Chapter 1

INTRODUCTION

1.1 Motivation

Blockchain technology has grown significantly beyond its initial use in cryptocurrency, now impacting many areas such as healthcare, supply chain management, e-voting, and agriculture. A major challenge in these systems is achieving high transaction throughput—the rate at which transactions are confirmed and added to the blockchain—while maintaining decentralization, security, and scalability. Although recent advances in consensus mechanisms have pushed throughput limits higher, often reaching thousands or even tens of thousands of transactions per second, the efficiency of the underlying peer-to-peer (P2P) network still limits real-world performance. Blockchain P2P networks usually form by new peers connecting to random existing nodes. Also, since peer selection is random, messages can take longer paths by jumping between distant nodes instead of closer ones, increasing convergence time and slowing propagation.

1.2 Problem Statement

Several methods have been proposed to use geographical proximity to connect peers that are nearer to each other. While this nearest neighbour connectivity (NNC) approach helps to reduce the latency in the network, it creates network imbalance and leads to formation of hub nodes. The hub nodes connect to an large number of other peers becoming single points of failure, and leads to decentralization. Therefore, an ideal peer

connection protocol necessitates a global view of the current P2P network state to optimize peer connectivity that ensures fast message propagation, balance peer connections while maintaining decentralization

1.3 Objectives

The objective of this project is to design an optimized peer connectivity approach for blockchain P2P networks that improves message propagation and reduces convergence time.By using a Digital Twin model of the P2P network, the goal is to make better connections, reduce network latency, and avoid the formation of overloaded hub nodes.

1.4 Scope of the Study

This study aims to develop an optimized peer-to-peer (P2P) connectivity system for blockchain networks. The scope of the study includes:

1. Analysis of Existing P2P Network Topologies

- Review of current P2P connectivity methods, especially the Nearest Neighbor Connectivity (NNC) approach.

2.Design and Development of a Digital Twin–Based Connectivity Framework

- Developing a connectivity framework that uses a virtual representation of the P2P network.

3. Improvement of Convergence Time

- Developing strategies to enhance the convergence time for newly joining peers.

- Comparing the performance of the new system with the NNC and random connection approaches to validate improvements.

4. Evaluation and Testing

- Implementing the proposed peer connectivity system and message propagation mechanisms in a simulation environment.
- Conducting performance evaluations to assess improvements in transaction throughput, convergence time, and message propagation.

5. Limitations

- The study will be limited to the analysis and testing within simulated environments, with potential for real-world application to be considered in future research.

This study does not encompass the development of new consensus algorithms or alterations to the blockchain protocol itself, but rather focuses on optimizing the P2P network layer to support existing blockchain infrastructures.

1.5 Thesis Outline

This thesis is organized into six chapters. This structured approach ensures a thorough exploration from conceptual design to practical application and evaluation.

- Chapter 1 - Introduction: This chapter provides an overview of the project's background, problem statement, objectives, scope, and structure.
- Chapter 2 - Literature Review: This chapter reviews existing research and methods in optimizing the p2p network of blockchain.
- Chapter 3 - System Design: This chapter elaborates upon the design of the proposed system "TwinPeer".

- Chapter 4 - Implementation Details: This chapter describes the implementation details of the proposed TwinPeer system.
- Chapter 5 - Experimental Results: This chapter presents the experimental results of the proposed TwinPeer system.
- Chapter 6 - Conclusion : This chapter summarizes the key findings, draws conclusions.

Chapter 2

LITERATURE REVIEW

2.1 Background and Motivation

Blockchain architectures are typically composed of three foundational layers: the *Application layer*, the *Consensus layer*, and the *Peer-to-Peer (P2P) Network layer*. The Application layer facilitates the deployment of cryptocurrencies and decentralized applications (dApps) through smart contracts. The Consensus layer ensures that network participants reach agreement on the validity of transactions and block ordering, using mechanisms such as Proof-of-Work (PoW) or Proof-of-Stake (PoS). Beneath these, the P2P Network layer acts as the communication backbone, handling peer discovery and disseminating transactions, blocks, and consensus messages, typically using gossip-based protocols.

A key limiting factor is the P2P network layer, which has become a significant bottleneck for achieving high transaction throughput in decentralized environments [10]. In blockchain networks, new peers typically establish connections with a small, fixed number of randomly selected existing peers — for example, 8 in Bitcoin [12]. Message propagation in these networks relies on gossip protocols, which while robust, generate substantial message redundancy and inefficient dissemination paths. The random nature of peer connections introduces unpredictable message propagation delays and increases convergence times, directly impacting network responsiveness [1].

These propagation delays become especially critical in latency-sensitive domains such

as Decentralized Finance (DeFi) and Internet of Things (IoT) applications, where even slight differences in propagation speed can confer significant advantages to certain users or degrade the overall system performance. Additionally, network congestion during peak transaction periods, combined with the geographically distributed nature of blockchain nodes, exacerbates latency issues, especially when peer selection lacks latency awareness. Highly decentralized systems, while more resilient, involve a greater number of validators and longer propagation paths, increasing the time required for network-wide consensus. As transaction volumes grow, scalability bottlenecks in the P2P messaging infrastructure further restrict a blockchain's ability to leverage fast consensus algorithms, making the efficiency of the P2P layer a decisive factor in real-world throughput.

2.1.1 Digital Twin Concepts for Network Management

In the context of distributed and dynamic P2P networks like blockchains, recent developments in Digital Twin technologies offer promising opportunities for improving network observability and performance management. A Digital Twin is a virtual, continuously synchronized representation of a physical or logical system, maintaining real-time data on system state, topology, connectivity, and performance metrics. Applied to blockchain networks, a Digital Twin could track peer connections, network latency, geographic distribution, and other network characteristics, providing a comprehensive and up-to-date view of the network's structure and behavior.

This global visibility is especially valuable in decentralized and dynamic environments where network topologies constantly evolve due to peer churn and variable traffic loads. Digital Twins can assist in identifying propagation bottlenecks, load imbalances, and suboptimal peer connections, supporting more effective network optimization strategies. Additionally, their ability to simulate network behavior and evaluate alternative peer selection or clustering strategies virtually, before applying them to the live network, offers a safe and proactive means of enhancing P2P network performance. While widely used in industrial IoT and large-scale distributed systems, the application of Digital Twin concepts to blockchain P2P networks remains an emerging research area with significant potential.

2.2 Related Work

This chapter presents related work on optimizing the p2p network latency of the blockchain network.

2.2.1 Non-Clustered Approaches for Blockchain P2P Networks

Maya et al. [1] conducted a survey that identifies the P2P network of blockchain as a bottleneck for transaction throughput. This work proposed a technique to optimize information exchange between nodes to expedite block propagation for faster transaction handling with reduced delay. Zhou et al. [7] proposed the MERCURY protocol, which employs a virtual coordinate system (VCS) and an early outburst strategy to reduce propagation latency by up to 56%, maintaining low bandwidth usage and strong resilience to attacks. However, dependency on a secure VCS may increase complexity in dynamic or constrained networks. LECast [27], proposed by Luo et al., uses a shortest-path broadcast tree to reduce energy consumption in UAV blockchain networks. Zheng et al. later introduced DHBN [28], which applies Minimum Latency Broadcast Trees (MLBT) to support fast, dynamic environments. While both methods target blockchain efficiency, they differ in goals and design. Valko et al. [29] used reinforcement learning to reorder neighbor nodes based on latency, improving propagation speed. However, this method requires costly training and frequent retraining, limiting its scalability.

Andrew et al. [12] analyzed transaction latency in private ethereum blockchain-based IoT networks utilizing PoW and PoS consensus mechanisms. They described the transaction lifecycle into three components: transaction processing time, mining time (for PoW), and block addition time. The authors highlight latency concerns within the P2P network in both contexts.

In contrast, Varun et al.[17] suggested an architecture based on Software-Defined Networking (SDN) to separate the control and data planes, enabling centralized control for agile network reconfiguration. These approaches highlight different strategies to enhance block propagation efficiency but overlook the impact of random topology on P2P network latency. Yifan et al. [3] propose Perigee, an intelligent algorithm designed to optimize

block delivery speed in blockchain networks by leveraging geographical location. Perigee autonomously learns and adapts to diverse network setups, with each node assessing block reception speed from neighbors to determine whether to maintain current connections or seek better ones. This approach burdens the nodes with latency tracking and dynamic topology reconfiguration. Both these approaches utilize geographical proximity to connect nodes efficiently.

2.2.2 Clustering approaches for Blockchain P2P Networks

Clustering techniques optimize network performance by organizing nodes into groups based on similar characteristics, reducing latency, message redundancy, and improving communication efficiency.

Building on clustering principles, Ke Zhang et al.[19] proposed a clustering-based node grouping algorithm using a two-stage attention decoder, assigning leader nodes to manage follower clusters. It improved Ethereum communication efficiency by 76.1%, though its computational overhead limits use in resource-constrained environments. Ran et al. [20] proposed DRL-DBSCAN, a Deep Reinforcement Learning-enhanced clustering protocol that dynamically selects leaders based on communication delay. It boosts scalability and throughput via Gossip-based inter-cluster consensus but remains vulnerable to malicious node attacks.

Hao et al. [30] introduced BlockP2P-EP, which uses K-Means clustering and a hierarchical topology to boost transmission speed and reliability in blockchain networks.

FRING [10] is a geography-based P2P overlay network aimed to enhance broadcast message efficiency and convergence time. FRING organizes nodes hierarchically into rings, with lower-level rings fully connected based on geographical proximity. Representatives from each ring form higher-level rings recursively until a single global ring remains at the top level. However, this approach's limitation lies in its highly restricted and attack-vulnerable topology. Hai et al. [21] proposed a proximity-based clustering algorithm for blockchain networks, using seed nodes to ensure efficient message forwarding with fewer hops. It reduces delays and block failures but struggles with scalability, maintenance overhead, and vulnerability to node failures. Yamina et al. [43] proposed ClusterChain, a framework where

multiple nodes locally execute clustering models and synchronize the optimal solution. The network architecture includes a manager node and miner nodes to collaboratively perform clustering. Zola et al. [33] employed clustering to separate malicious and benign blockchain activity, followed by ensemble classifiers like Random Forest and AdaBoost. Their model achieved a high accuracy of 99.68% in detecting threats within the Bitcoin network.

Fadhil et al. introduced two clustering protocols, namely BCBSN [24] and LBC [25], where clustering is performed based on the geographical positioning of the nodes.Owenson et al. [26] developed a clustering protocol known as BCBPT, which leverages ping delay for cluster formation. Experimental analysis demonstrated that BCBPT outperforms BCBSN and LBC in minimizing propagation delays between neighboring nodes within a cluster. Zhao et al.[31] proposed a lightweight Bitcoin topology discovery algorithm using probe nodes and node-layer clustering, achieving 95% accuracy with 72% less communication overhead compared to Bitnodes. Cheng et al.[32] proposed FPSblo-EP, a hierarchical model using geographic sampling and node degrees to enhance blockchain routing. It reduces latency by 22%, redundancy by 26%, and stretch by 28%, improving scalability and reliability. Ben Othmen et al.[42] proposed RC-PBFT, leveraging random clustering to perform localized PBFT consensus and reduce communication overhead. However, the approach lacks optimized cluster selection and offers limited resilience to Byzantine faults.

2.2.3 Digital Twin related work

Liu et al. [14] reviewed blockchain-based DT implementations in smart manufacturing and logistics, highlighting how blockchain enhances security and traceability by providing tamper-proof records. Similarly, Sadri et al. [15] examined the integration of blockchain and digital twins in smart building management, focusing on secure data exchange for asset life-cycle management. Singh et al. [16] proposed a blockchain solution using Ethereum and IPFS to secure digital twin data sharing, ensuring data integrity and trust. These approaches integrate blockchain and digital twin for different purposes but does not consider the usage of modelling peer to peer network of blockchains as digital twins that helps to enhance network efficiency by optimizing peer connections, especially for large-scale systems.

Chapter 3

SYSTEM DESIGN

3.1 System Overview

This work presents TwinPeer, a Digital Twin–driven peer connectivity optimization framework designed to enhance propagation efficiency, decentralization, and scalability in blockchain P2P networks. At its core, TwinPeer employs a Digital Twin for Blockchain P2P Networks (DT-BPN), a continuously updated virtual replica of the physical blockchain P2P network. In conventional blockchain systems, bootstrap nodes typically provide new peers with a list of active nodes (via IP addresses) to help them connect to the P2P network. Building on this practice, a DT-BPN is developed to virtually store and maintain the real time information of the peers including geo-location (latitude and longitude), their IP address and connectivity with other peers. DT-BPN provides an up-to-date global view of the P2P network state that forms the foundation for informed, latency-aware, and balanced peer management decisions while overcoming the inherent visibility limitations of decentralized P2P network.

By leveraging DTPN, the proposed ”Proximal and Degree Balanced Connectivity Algorithm (PDBCA),” recommends peers to based on their proximity and current degree of connections. DTPN helps PDBCA take informed decisions by providing the real time network state when recommending peers to achieve improved convergence time and prevents the formation of hub nodes, which would otherwise compromise network decentralization and resilience. The design of the proposed system is illustrated in Fig 3.1, and explained in

the following steps as enumerated in the figure.

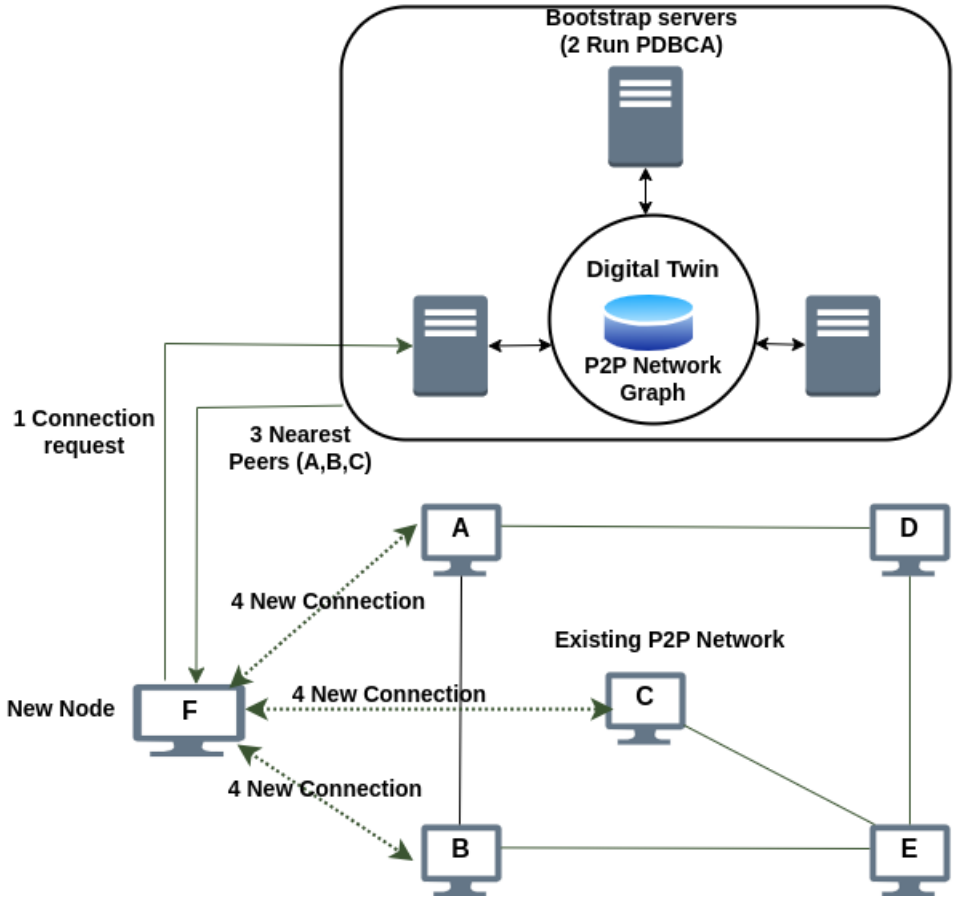


Figure 3.1: For a new node F, PDBCA returns the IP addresses of the nearest peers A,B,C. Connections are established with peers A,B,C.

Every bootstrap node maintains the virtual representation of the evolving P2P network (DTPN) in the form of a graph, DTPN_Graph with vertices as peers and edge costs as geographical distances between the peers. Upon receiving a request for a list of active peers from a new peer (1), bootstrap node learns the latitude and longitude information of the new peer based on its IP address. It then runs the PDBCA to finds a minimum set of active neighbor peers (r_{min}), by taking into consideration the proximity as well as the node degrees of existing peers for connection establishment for the new peer (2). r_{min} refers to the minimum number of connections a new peer makes to join the network. PDBCA balances the network and avoids the hub nodes problem by dynamically increasing the maximum degree of the overall network r_{max} (the maximum of all peer degrees in the P2P network), one step at a time as the new peers join the network.

PDBCA sends this set of neighbour peers, and their IP addresses to the new peer to connect to P2P network (3). The new peer then establishes connections with all the suggested peers (4). If a peer has not responded to the connection request, the new peer can request for other peers so that r_{min} connections are established. The bootstrap node now updates the DTPN with the new peer and their connections to the existing peers.

3.1.1 Proximal and Degree Balanced Connection Algorithm (PDBCA)

PDBCA shown in Algorithm 1 is designed to select an optimal set of peers for a new incoming peer while maintaining a balance between geographical proximity and the degree of the nodes. PDBCA follows a greedy approach that score the peers based on their relative distance (proximity) to the new peer and their node degree, sorts them, and selects r_{min} peers iteratively from the sorted list. PDBCA uses two parameters α and β as weight factors for the proximity and degree respectively when calculating the score of each peer.

Algorithm 1 Proximal and Degree Balanced Connection Algorithm (PDBCA)

Input: NewNode, r_{min}, α, β, Graph G (from DTPN)
Output: SelectedNodeIPs
Initialize r_{max} // The maximum allowable degree in the P2P network.
Initialize an empty list SelectedNodes.
Calculate Haversine distance between NewNode_N and all existing nodes.
Compute Peer_Scores for all existing nodes using Eq 1, 2 and 3.
Sort all the peers by P_$Score$ value in ascending order to get list D.
while length(SelectedNodes) $< r_{\text{min}}$ **do**
    for each node in D **do**
        if degree(node) $\leq r_{\text{max}}$ **and** node $\notin$ SelectedNodes **then**
            Add node to SelectedNodes
            if length(SelectedNodes) >= r_{min} **then**
                break
            end if
        end if
    end for
    if length(SelectedNodes) $< r_{\text{min}}$ **then**
        $r_{\text{max}} \leftarrow r_{\text{max}} + 1$
    end if
end while
return SelectedNodeIPs

The algorithm starts by calculating the Haversine Distance (HD) between the new peer and all existing nodes in the graph. Haversine distance , is calculated using the latitude and

longitude information derived from the IP addresses of peers, given by equation 3.1:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \tag{3.1}$$

where: a is squared half-chord length between the two points on the unit sphere; ϕ_1 and ϕ_2 are the latitudes of the two points in radians; λ_1 and λ_2 are the longitudes of the two points in radians; $\Delta\phi = \phi_2 - \phi_1$; $\Delta\lambda = \lambda_2 - \lambda_1$ The central angle c between the two peers is given by $c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$. Finally, the distance d between the two peers is found by $d = R \cdot c$, where, R is the Earth's radius.

PDBCA calculates two scores for each peer - Proximity Score (Pr_Score) and Degree Score (D_Score), given by 3.2 and 3.3 respectively. While Pr_Score contributes to the selection of geographically closest peer, the D_Score contributes to minimize the imbalance of the peer degrees in the P2P network. Finally the overall score for each peer $Peer_Score$ is calculated as the weighted combination of these two scores, where α and β are weighting factors that control the influence of geographical proximity and degree respectively, while selecting neighbors for the new peer.

$$\text{Pr_Score} = \frac{\text{HD from the new peer}}{\text{Maximum HD to all nodes from the new peer}} \tag{3.2}$$

$$\text{D_Score} = \frac{\text{Degree of the peer}}{\text{Maximum degree } (r_{\text{max}}) \text{ in the P2P network}} \tag{3.3}$$

$$\text{Peer_Score} = \alpha \times \text{Pr_score} + \beta \times \text{D_score}; \alpha + \beta = 1 \tag{3.4}$$

All the peers are then sorted based on their $Peer_Score$ values, with lower scores indicating better candidates for selection. The algorithm selects r_{min} peers from the sorted list. If a node's degree is less than or equal to r_{max} and it has not already been selected, it is added to the list of selected nodes. If fewer than r_{min} nodes are selected after one pass through the sorted list, r_{max} is incremented, and the process is repeated. The increase in value of r_{max} is typically done when all the peers have reached the maximum allowable degree and there are no peers with current r_{max} value. This iterative process ensures that

the new peer connects to peers that are both geographically close and balanced in terms of network degree, while avoiding the formation of hub nodes. The final result is a set of selected node IPs that meet the required conditions.

3.1.2 Comparison of NNC and PDBCA Algorithms

The Nearest Neighbor Connection (NNC) approach identifies the $r_{\min}$ geographically closest peers for every new peer requesting to join the P2P network. It selects the nearest peers based solely on geographical proximity without considering the degree imbalance that may occur in the newtork. This approach, while simple, can lead to the formation of hub nodes. This scoring mechanism allows PDBCA to avoid connecting the new peer to existing peers that are already highly connected (i.e., hub nodes), even if they are geographically closer. Instead, PDBCA favors nodes with a lower degree, promoting a more balanced and resilient network structure.

Fig. 3.2 shows the behaviour of PDBCA and NNC approaches for a sample P2P network graph of 10 nodes, and the numbers on the edges represent the distances between the peers. Assuming that each peer connects to atleast two nearest neighbors, NNC approach results in the formation of two hub nodes C, F that have a degree of 5 each. However, in the case of PDBCA, the degree of C and F are controlled as they grow beyond 2. The nodes D, E, I establish connections with G, H and J respectively even though these peers are not the closest compared to the peers C,F, resulting in better degree balance of the network.

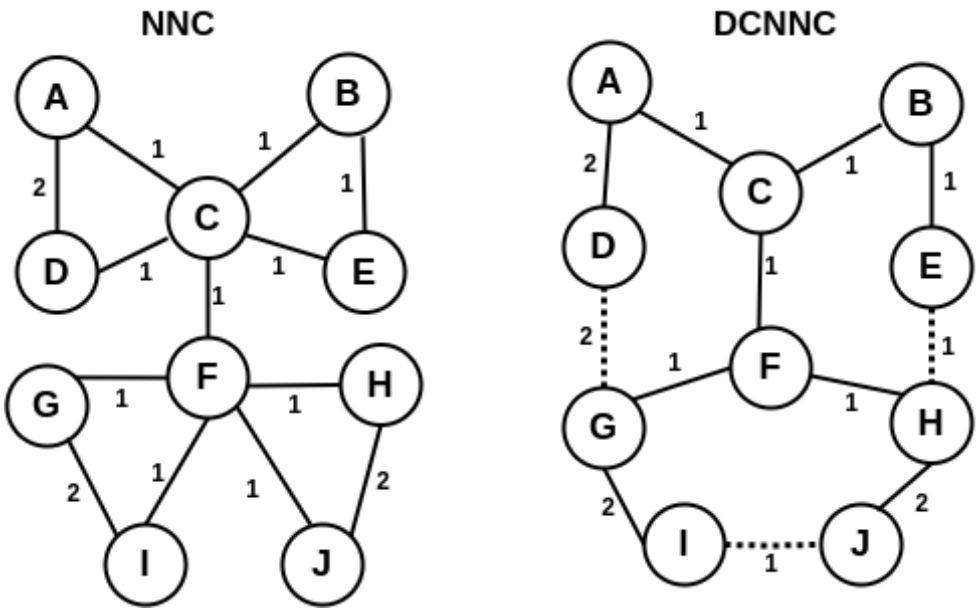


Figure 3.2: Sample topology formed based on NNC and PDBCA algorithms. Nodes degrees are evenly distributed and hub nodes C and F in NNC are avoided in PDBCA.

3.1.3 Time Complexity of PDBCA

Assuming the number of existing nodes (peers) in the DTPN Graph is N at a given point in time. Calculation of the Haversine distance takes constant ($O(1)$) time. The Haversine distance calculation between the new node and existing nodes takes $O(N)$ time. Calculating the scores for each node based on both distance and degree takes $O(N)$ time. Sorting the scores list D takes $O(N \log N)$ time. Selecting the minimum number of neighbors required takes $O(N)$. From all the above steps, the total time complexity of the PDBCA algorithm is $O(N \log N)$ time. The complexity of PDBCA become increases as the size of the network (N) increases.

Chapter 4

IMPLEMENTATION DETAILS

In the TwinPeer framework, Neo4j serves as the back-end system to build and maintain the Digital Twin for Blockchain P2P Networks (DT-BPN). Neo4j is a widely used graph database that helps maintain persistent, real-time, and queryable virtual replica of the underlying blockchain P2P network. It continuously records and updates essential peer information including IP addresses, geographical coordinates (latitude and longitude) and dynamic peer-to-peer connectivity as nodes join or leave the network. The interaction with Neo4j is performed using Cypher, its expressive graph query language, which supports a variety of operations for TwinPeer,including retrieving peer connection details, and analyzing the evolving network structure. Through continuous synchronization with real-time network changes, Neo4j ensures that the DT-BPN accurately reflects the current state of the physical P2P network at all times.

Complementing this, NetworkX, a Python-based graph analysis library, is used locally on the Bootstrap server to build, manipulate, and analyze the operational Blockchain P2P Network Graph (BPNG). In this in-memory representation of BPNG, peers are modeled as nodes with associated properties such as IP address, latitude, longitude, while peer connections are modeled as weighted edges based on geographical proximity computed using the haversine distance.

BPNG graph constructed and maintained using NetworkX, serves as the computational platform to execute Proximal and Degree Balanced Connection Algorithm (PDBCA), enabling proximity-aware, load-balanced peer selection. BPNG graph once generated is synchronized with the Neo4j database. Nodes representing peers are inserted or updated

in Neo4j using Cypher merge queries, while peer connections are stored as relationships annotated with their respective haversine distances. For example, the following merge queries are used to add peers and record peer-to-peer connections respectively.

```
MERGE (n:IPAddress {ip: $ip})
SET n.latitude = $lat,
   n.longitude = $lon
```

```
MATCH (n1:IPAddress {ip: $node1})
MATCH (n2:IPAddress {ip: $node2})
MERGE (n1)-[r:CONNECTS]->(n2)
SET r.distance = $distance
```

To retrieve the P2P network information, bootstrap servers authenticate and query Neo4j via Cypher, extracting both peer and relationship data. This structured result is converted back into a NetworkX graph for further processing, enabling efficient addition of new peers, removal of inactive nodes and reorganization of connections. A sample Cypher query to retrieve peer relationships is:

```
MATCH (n:IPAddress)-[r:CONNECTS]->(m:IPAddress)
RETURN n.ip, n.latitude, n.longitude,
     r.distance,
     m.ip, m.latitude, m.longitude
```

This bi-directional synchronization process between NetworkX and Neo4j ensures that TwinPeer maintains a reliable, real-time digital twin that reflects the operational P2P network, enabling decentralized peer management with global visibility and fine-grained control.

Chapter 5

EXPERIMENTAL RESULTS

P2P networks are generated using NetworkX python library using GeoIP database provided by MaxMind which contains information about different nodes and their IP addresses that are randomly spread across different regions of the world. For every IP address, the associated latitude and longitude coordinates are obtained. This information is used to to generate graphs that reflect the behaviours of PDBCA, NNC and Random approaches. These algorithms are developed using Python 3.10.12 and run on system with Intel 8 core processor with 8GB RAM.

Experiments were performed on graphs of different sizes including 1000, 2000, 5000 and 10000 and 20000 nodes from the GeoIP database. The geographical distances between a pairs of nodes (edges) are calculated based on the Haversine distance using their coordinates. The propagation delay of 5ms is used for a geographical distance of 1000km based on the assumption that in a fibre optic cable, the propagation delay is observed to be 5 microseconds for 1km. The values of α and β are both set to 0.5 in PDBCA to ensure that peers are selected by giving equal weights to both proximity and degree. The r_{min} value (minimum number of connections to other nodes) and r_{max} value (maximum degree of the network) are set to 5 and 20 respectively. r_{max} is set sufficiently large compared to r_{min} to ensure PDBCA behaves similar to NNC initially, but gradually controls the overall maximum degree of the network as the network grows in size.

5.0.1 Comparison of Average Degree

The average degree of a network is calculated as the sum of all degrees of the nodes over the total number of nodes and it represents the overall balance of the P2P network. Fig 5.1 shows the average degree of P2P networks when generated using PDBCA, NNC and Random approaches. Since each node maintains a minimum number of connections, lower average degree means the network is balanced while high average degree means the network is imbalanced resulting in the formation of hub nodes. The PDBCA has less average degree compared to NNC and Random approaches, indicating that network is balanced. It can be clearly observed that the NNC has high average degree of all three approaches making the network highly imbalanced.

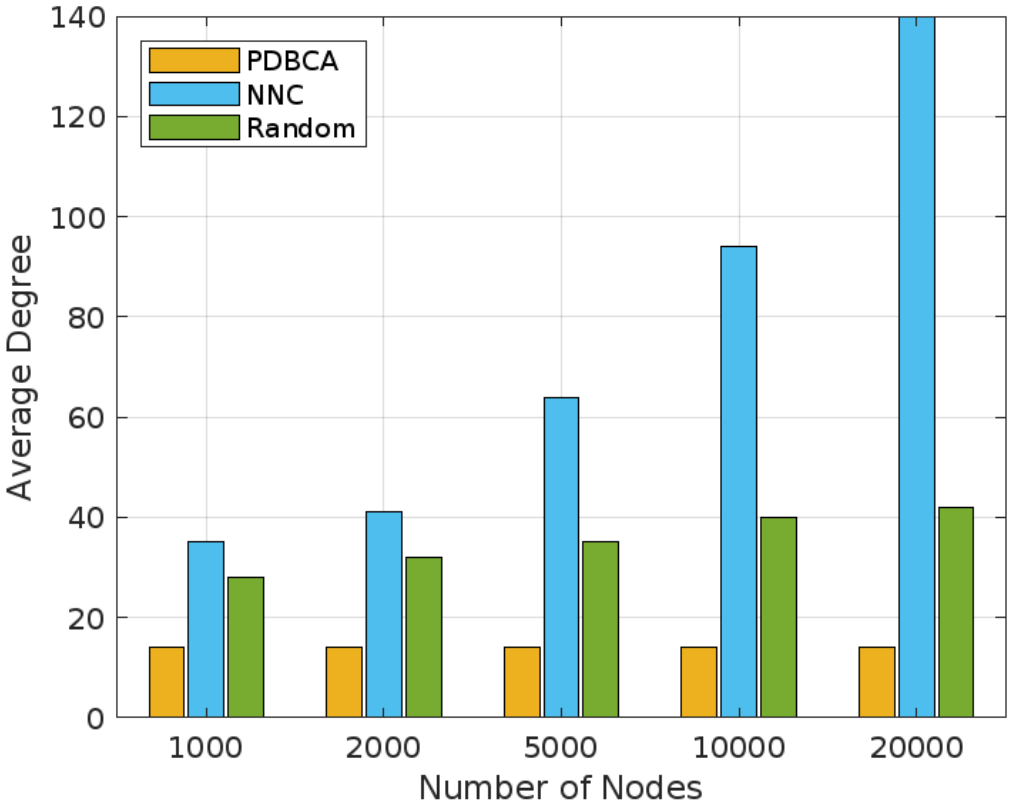


Figure 5.1: Comparison of average degree for different network sizes

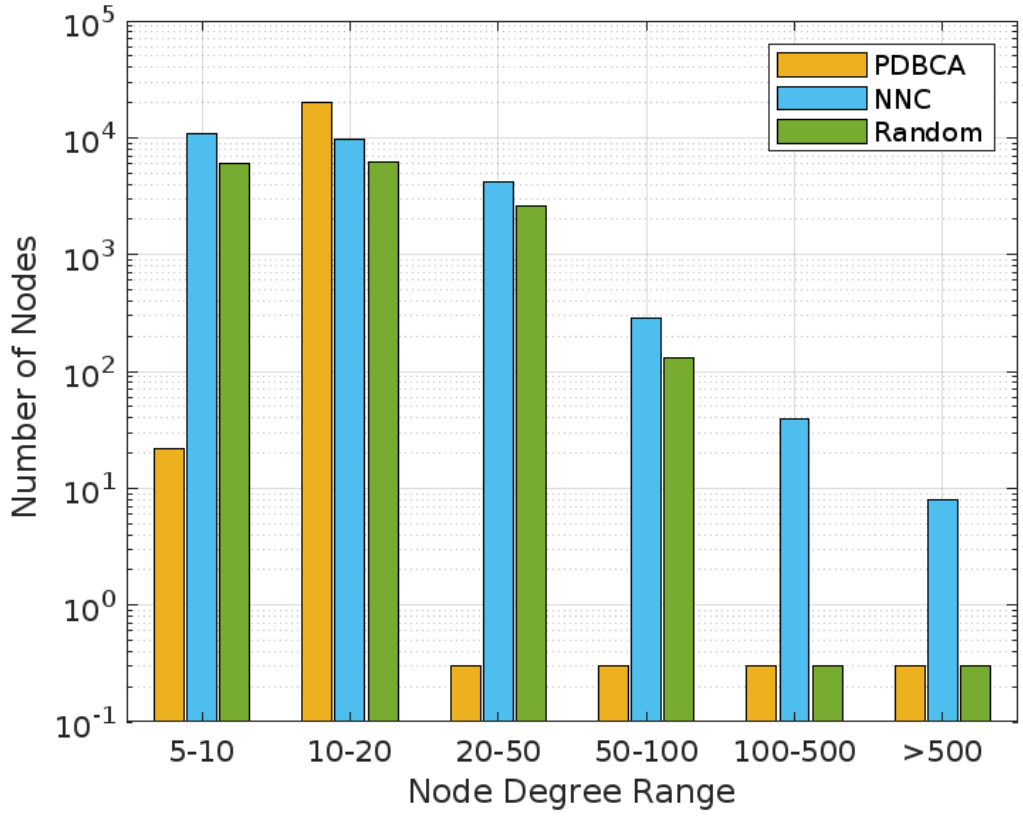


Figure 5.2: Comparison of degree distribution for 20k node network

Fig 5.2 shows the degree distribution for a network size of 20k nodes. All the nodes have a degree between 5 and 20 in PDBCA, balancing the entire network. In both NNC and Random approaches (where there is no control over $r_{\max}$), a significantly higher number of nodes have nodes with degrees in the range of 20-100. Around 300 nodes of NNC and 140 nodes in Random approaches have degrees between 50-100 respectively. Further, only in NNC approach it is observed that there are 40 nodes in 100-500 range and 8 nodes whose degree is above 500 with some nodes reaching node degree as high as 600. Such high degree nodes are prone to targeted attacks and leads to decentralization of the overall P2P network.

5.0.2 Convergence Time comparison

Convergence time refers to the total time a message initiated from a source node takes to reach all nodes in the network. Fig 5.3 shows that the average convergence time of all three approaches across various network sizes. PDBCA is better compared to the NCC, and the

Random approach results in a higher convergence time among all of them. PDBCA balances the network connections by connecting to proximal nodes and when required connects to far away nodes to balance the network to establish minimum number of connections.

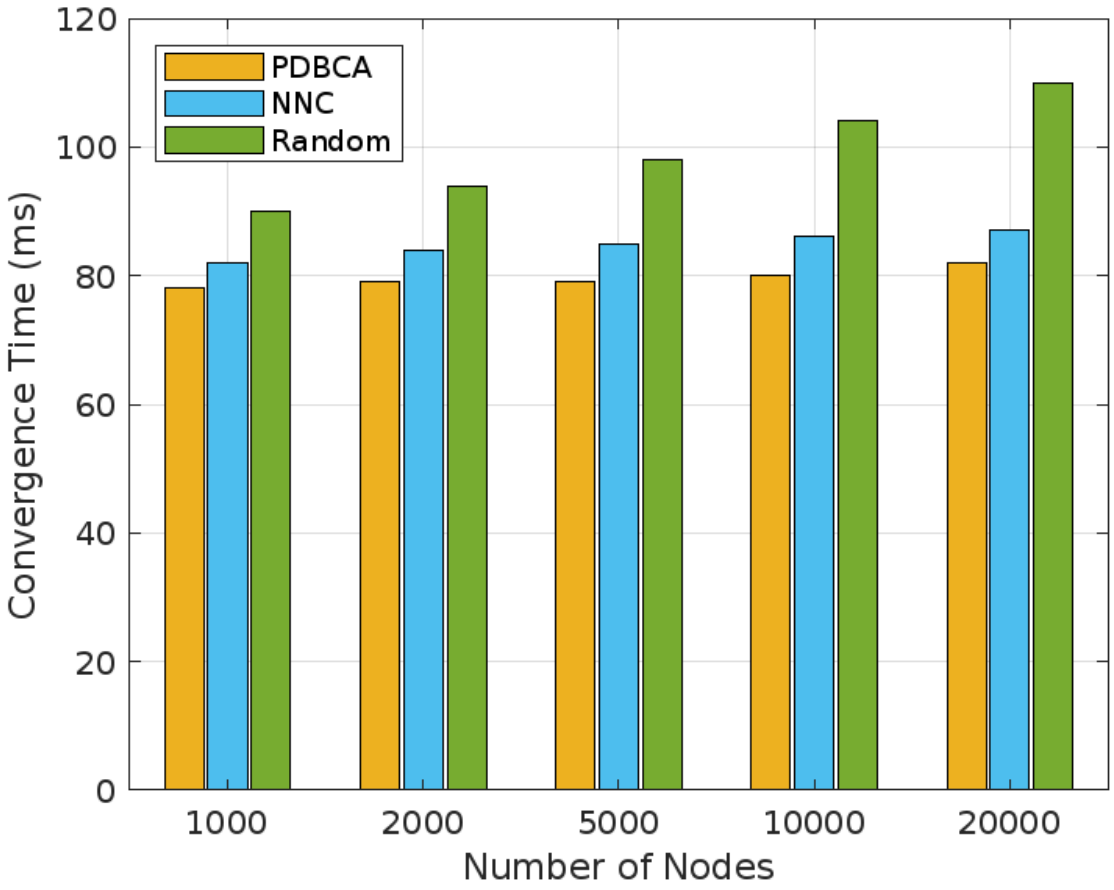


Figure 5.3: Comparison of Convergence time

An improved performance of PDBCA compared to NNC is observed because it connects to nearby neighbors as well as some far away nodes in a controlled manner. Connecting to far away nodes helps the message to reach all the nodes quickly, thereby improving the overall convergence time. Further, there is an increase in the number of messages that are being forwarded which helps PDBCA achieve quick convergence time compared to NNC. This performance enhancement compared to NNC while effectively mitigating the issue of hub nodes in the P2P network proves the overall effectiveness of proposed PDBCA.

5.0.3 Comparison of Message complexity

Message complexity refers to the total number of messages that are generated when a message initiated from source reaches all nodes in the network following the gossip protocol. Fig 5.4 shows the message complexity of all three approaches for various network sizes. PDBCA and Random approaches generate higher number of messages compared to NNC. Because the hub nodes connects to large number of other nodes, a message is sent to large number of peers from hub nodes only once reducing the overall message redundancy. However, hub nodes leads to centralization of the network and creates the overall network imbalance.

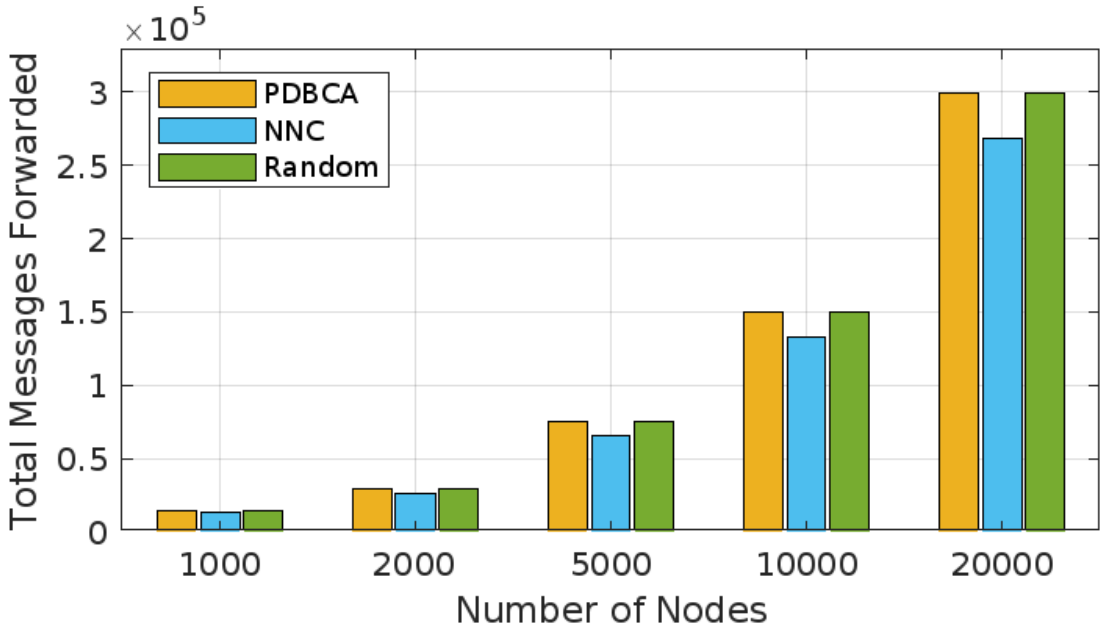


Figure 5.4: Comparison of Number of Messages Forwarded

Chapter 6

CONCLUSION

This paper introduces 'TwinPeer', a Digital Twin–based framework to optimize peer connectivity in blockchain P2P networks. DTPN leverages the 'Proximal and Degree Balanced Connectivity Algorithm' (PDBCA) algorithm that strategically connects peers based on geographical proximity as well as degree of the peers. PDBCA algorithm ensures that all nodes degrees are evenly distributed thereby preventing the formation of hub nodes. The experimental results prove that PDBCA achieves similar or better convergence compared to NNC and Random approaches. Future work involves addressing the challenges listed in Discussion section and exploring the concept of clustering to achieve greater reductions in message redundancy.

Bibliography

[1] Maya Dotan, Yvonne-Anne Pignolet,Steffan Schmid,Saar Tochner,Aviv Zohar " Survey on Blockchain Networking: Context, State-of-the-Art, Challenges", Proc. *ACM Computing Surveys, Vol. 54, No. 5, Article 107. Publication date: May 2021.*

[2] Lihao Zhang,Taotao Wang,Soung Chang Liew,"Speeding up block propagation in Bitcoin network: Uncoded and coded designs" . Proc. *Computer Networks 206 (2022) 108791*

[3] Yifan Mao, Soubhik Deb,Shaileshh Bojja Venkatakrishnan,Sreeram Kannan,Kannan Srinivasan,"Perigee: Efficient Peer-to-Peer Network Design for Blockchains", Proc. *In Symposium on Principles of Distributed Computing (PODC'20), August 3–7, 2020, Virtual Event, Italy.ACM, New York, NY, USA,*

[4] https://chainspect.app/dashboard?order=desc&sort=theorTps.

[5] Weizhao Tang, Lucianna Kiffer, Giulia Fanti, and Ari Juels. 2023. "Strategic Latency Reduction in Blockchain Peer-to-Peer Networks." Proc. *ACM Meas. Anal. Comput. Syst. 7, 2, Article 32 (June 2023)*

[6] B. Xue, Y. Mao, S. B. Venkatakrishnan and S. Kannan, "Goldfish: Peer Selection using Matrix Completion in Unstructured P2P Network," 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Dubai, United Arab Emirates, 2023, pp. 1-9.

[7] Mingxun Zhou, Liyi Zeng, Yilin Han, Peilun Li, Fan Long, Dong Zhou, Ivan Beschastnikh, Ming Wu, "Mercury: Fast transaction broadcast in high performance

blockchain systems", Proc. *IEEE INFOCOM 2023-IEEE Conference on Computer Communications, pp. 1-10, 2023, IEEE.*

[8] Kaushik Ayinala,Baek-Young Choi,Sejun Song,"PiChu: Accelerating Block Broadcasting in Blockchain Networks with Pipelining and Chunking", Proc. *2020 IEEE International Conference on Blockchain (Blockchain)*

[9] Xuan Chen,Kien Nguyen,Hiroo Sekiya, "On the Latency Performance in Private Blockchain Networks" , Proc. *IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 19, 1 OCTOBER 2022*

[10] Haoran Qiu, Tao Ji, Shixiong Zhao, , Xusheng Chen,Ji Qi,Heming Cui, "FRing: A Geography-Based P2P Overlay Network for Fast and Robust Blockchain Systems" Proc. *IEEE Transactions On Services Computing, VOL. 16, NO. 3, MAY/JUNE 2023.*

[11] 'Maxmind database', Poese, Ingmar, et al. "IP geolocation databases: Unreliable?." ACM SIGCOMM Computer Communication Review 41.2 (2011): 53-56.

[12] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2015. Discovering bitcoin's public topology and influential nodes.

[13] Neo4j Graph Database. *https://www.maxmind.com/en/solutions/geoip2-enterprise-product-suite/an*

[14] Liu, Jiongbin, et al. "Blockchain-based digital twin for supply chain management: a literature review and future research directions." arXiv preprint arXiv:2202.03966 (2022).

[15] Sadri, Habib, et al. "Integration of blockchain and digital twins in the smart built environment adopting disruptive technologies—A systematic review." Sustainability 15.4 (2023): 3713.

[16] A. Onwubiko, R. Singh, S. Awan, Z. Pervez, and N. Ramzan, "Enabling trust and security in digital twin management: A blockchain-based approach with Ethereum and IPFS," *Sensors*, vol. 23, no. 14, p. 6641, 2023.

[17] Varun Deshpande,Hakim Badis,Laurent George, ”Efficient topology control of blockchain peer to peer network based on SDN paradigm”. Proc. *Peer-to-Peer Networking and Applications (2022) 15:267–289*

[18] Chopde, Nitin R., and Mangesh Nichat. ”Landmark based shortest path detection by using A* and Haversine formula.” International Journal of Innovative Research in Computer and Communication Engineering 1.2 (2013): 298-302

[19] Ke Zhang, Zhenwen Peng, Ruixing Zong, Qiong Wang, Xiong Xiao, Zhuo Tang, ”Communication Optimization in Blockchain Peer-to-Peer Networks”, Proc. *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 742-747, 2024, IEEE.*

[20] Peiyun Ran, Yipeng Ji, Mingsheng Liu, Peng Zhao, Shiyuan Yu, Yongjian Huang, Du Wang, Zakirul Alam Bhuiyan, Gang Li, ”Blockchain topology optimization based on node clustering”, Proc. *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1-6, 2023, IEEE.*

[21] Mo Hai, Wenchang Li, Haifeng Li, ”An efficient propagation algorithm based on node clustering and seed node selection in a blockchain network”, Proc. *Peer-to-Peer Networking and Applications, Vol. 18, No. 1, pp. 1-15, 2025, Springer.*

[22] Ling Ding, Chao Li, Di Jin, Shifei Ding, ”Survey of spectral clustering based on graph theory”, *Pattern Recognition*, Vol. 1, No. 1, pp. 110366, 2024, Elsevier.

[23] Guia, J., Soares, V. G., Bernardino, J. (2017). ”Graph databases: Neo4j analysis,” ICEIS (1), 351–356. Elsevier.

[24] Fadhil M, Owenson G, Adda M (2016) A bitcoin model for evaluation of clustering to improve propagation delay in bitcoin network. In: 2016 IEEE Intl conference on computational science and engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES). IEEE, pp 468–475

[25] Fadhil M, Owenson G, Adda M (2017) Locality based approach to improve propagation delay on the bitcoin peer-to-peer network. In: 2017 IFIP/IEEE Symposium on integrated network and service management (IM). IEEE, pp 556–559

[26] Owenson G, Adda M et al (2017) Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network. In: 2017 IEEE 37th International conference on distributed computing systems (ICDCS). IEEE, pp 2411–2416

[27] Luo H, Liu S, Xu S, Luo J (2023) Lecast: A low-energy-consumption broadcast protocol for uav blockchain networks. Drones 7(2):76

[28] Zheng R, Luo H, Sun G, Yu H (2024) Dhbn: An efficient broadcast protocol for blockchain networks in highly dynamic heterogeneous environment. In: 2024 IEEE Wireless communications and networking conference (WCNC). IEEE, pp 1–6

[29] Valko D, Kudenko D (2024) Sustainable broadcasting in blockchain network with reinforcement learning. arXiv:2407.15616

[30] Weifeng Hao, Jiajie Zeng, Xiaohai Dai, Jiang Xiao, Qiang-Sheng Hua, Hanhua Chen, et al., "Towards a trust-enhanced blockchain p2p topology for enabling fast and reliable broadcast", IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 904-917, 2020.

[31] Zhao, Z., Wang, J., Yang, M., Wang, H. (2025). An efficient Bitcoin network topology discovery algorithm for dynamic display. Blockchain: Research and Applications, 100260.

[32] Cheng, L., Tan, H., Li, X. et al. A hierarchical overlay network optimisation model for enhancing data transmission performance in blockchain systems. Sci Rep 14, 31900 (2024).

[33] F. Zola, M. Eguimendia, J. L. Bruse, and R. O. Urrutia, "Cascading machine learning to attack bitcoin anonymity," in 2019 IEEE International Conference on Blockchain (Blockchain). IEEE, 2019, pp. 10–17.

[34] von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z

[35] Galluccio, L., Michel, O., Comon, P., Hero III, A. O. (2012). Graph based k-means clustering. Signal Processing, 92(9), 1970-1984.

[36] Zhang, Wei, et al. "Graph degree linkage: Agglomerative clustering on a directed graph." Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12. Springer Berlin Heidelberg, 2012.

[37] He, Xiaofei, et al. "Laplacian regularized Gaussian mixture model for data clustering." IEEE transactions on knowledge and data engineering 23.9 (2010): 1406-1418.

[38] Maria, E., Budiman, E., Haviluddin, Taruk, M. (2020). Measure distance locating nearest public facilities using Haversine and Euclidean Methods. Journal of Physics: Conference Series, 1450(1), 012080.

[39] Hagberg, Aric, et al. "Exploring network structure, dynamics, and function using NetworkX." , Jan. 2008.

[40] Platt, E. L. Network science with Python and NetworkX quick start guide: explore and visualize network data effectively. Packt Publishing Ltd, 2019.

[41] Bitnodes, "Global Bitcoin nodes distribution," https://bitnodes.io/.

[42] R. Ben Othmen, W. Abbessi, S. Ouni, W. Badreddine, and G. Dequen, "Simulation of optimized cluster based PBFT blockchain validation process," in Proc. IEEE Symp. Comput. Commun. (ISCC), Jul. 2023, pp. 1317–1322.

[43] Khenfouci Yamina, Challal Yacine, and Hamdad Leila. Clusterchain: Decentralized and trustworthy clustering over blockchain. In 2021 International Conference on Networking and Advanced Systems (ICNAS), pages 1–6. IEEE, 2021.